

Splitting methods for Levitron Problems

Jürgen Geiser ^{*} Karl Felix Lüsrow [†]

January 10, 2012

Abstract

In this paper we describe splitting methods for solving Levitron, which is motivated to simulate magnetostatic traps of neutral atoms or ion traps. The idea is to levitate a magnetic spinning top in the air repelled by a base magnet.

The main problem is the stability of the reduced Hamiltonian, while it is not defined at the relative equilibrium. Here it is important to derive stable numerical schemes with high accuracy. For the numerical studies, we propose novel splitting schemes and analyze their behavior. We deal with a Verlet integrator and improve its accuracy with iterative and extrapolation ideas. Such a Hamiltonian splitting method, can be seen as geometric integrator and saves computational time while decoupling the full equation system.

Experiments based on the Levitron model are discussed.

Keywords splitting method, Verlet integrator, iterative and extrapolation methods, Levitron problem.

AMS subject classifications. 65M12, 65L06, 65P10.

1 Introduction

We are motivated to simulate a Levitron, which is a magnetic spinning top and can levitate in a magnetic field. The main problem of such a nonlinear problem is to achieve a stability for the calculation of the critical splint rate. While the stability of Levitrons are discussed in the work of [3] and their dynamics in [?], we concentrate on improving the standard time-integrator schemes for the reduced Hamiltonian systems. It is important to derive stable numerical schemes with high accuracy to compute the non-dissipative equation of motions. For the numerical studies, we propose novel splitting schemes and analyze their behavior. We deal with a standard Verlet integrator and improve its accuracy with

^{*}University of Greifswald, Institute of Physics, Felix-Hausdorff-Str. 6, D-17489 Greifswald, Germany, E-mail: juergen.geiser@uni-greifswald.de

[†]University of Greifswald, Institute of Physics, Felix-Hausdorff-Str. 6, D-17489 Greifswald, Germany, E-mail: tt-karl@freenet.de

iterative and extrapolation ideas. Such a Hamiltonian splitting method, can be seen as geometric integrator and saves computational time while decoupling the full equation system, see the splitting ideas in the overview article [1].

In the following we describe the reduced model of Gans [6] and an extension based on a novel idea of magnetic field of Dullin [3] for a disk.

1.1 Hamiltonian of Gans

In the paper, we deal with the following problem (reduced Hamiltonian):

$$H = \frac{1}{2} \left(p_1^2 + p_2^2 + p_3^2 + \frac{p_4^2}{a} + \frac{(p_5 - P - 6 \cos q_4)^2}{a \sin^2 q_4} + \frac{p_6^2}{c} \right) - M \left[\sin q_4 \left(\cos q_5 \frac{\partial \Psi}{\partial q_1} \sin q_5 \frac{\partial \Psi}{\partial q_2} \right) + \cos q_4 \frac{\partial \Psi}{\partial q_3} \right] + q_3 \quad (1)$$

The evolution of the dynamical variable $u(\mathbf{q}, \mathbf{p})$ (including \mathbf{q} and \mathbf{p} themselves) is given by the Poisson bracket,

$$\partial_t u(\mathbf{q}, \mathbf{p}) = \left(\frac{\partial u}{\partial \mathbf{q}} \cdot \frac{\partial H}{\partial \mathbf{p}} - \frac{\partial u}{\partial \mathbf{p}} \cdot \frac{\partial H}{\partial \mathbf{q}} \right) = (A + B)u(\mathbf{q}, \mathbf{p}). \quad (2)$$

For the non-separable Hamiltonian of (1), we have:

$$\dot{\mathbf{q}} = \frac{\partial H}{\partial \mathbf{p}}(\mathbf{p}, \mathbf{q}) = \left(p_1, p_2, p_3, \frac{p_4}{a}, \frac{(p_5 - p_6 \cos q_4)^2}{a \sin^2 q_4}, \frac{p_6(\cos^2 q_4 + (a/c) \sin^2 q_4) - p_5 \cos q_4}{a \sin^2 q_4} \right) \quad (3)$$

The same is given for:

$$\begin{aligned} \dot{\mathbf{p}} &= - \frac{\partial H}{\partial \mathbf{q}}(\mathbf{p}, \mathbf{q}) \\ &= \left(M \left(\sin q_4 \cos q_5 \frac{\partial^2 \Psi}{\partial q_1^2} + \cos q_4 \frac{\partial^2 \Psi}{\partial q_1 \partial q_3} \right), M \left(\sin q_4 \cos q_5 \frac{\partial^2 \Psi}{\partial q_2^2} + \cos q_4 \frac{\partial^2 \Psi}{\partial q_2 \partial q_3} \right), \right. \\ &\quad M \left(\sin q_4 \left(\sin q_5 \frac{\partial^2 \Psi}{\partial q_2 \partial q_3} + \cos q_5 \frac{\partial^2 \Psi}{\partial q_1 \partial q_3} \right) + \cos q_4 \frac{\partial^2 \Psi}{\partial q_3^2} \right) - 1, \\ &\quad M \left(\cos q_4 \left(\sin q_5 \frac{\partial \Psi}{\partial q_2} + \cos q_5 \frac{\partial \Psi}{\partial q_1} \right) - \sin q_4 \frac{\partial \Psi}{\partial q_3} \right) - \frac{p_6(p_5 - p_6 \cos q_4)}{a \sin q_4} - \frac{\cos q_4(p_5 - p_6 \cos q_4)^2}{a \sin^3 q_4}, \\ &\quad \left. M \left(\sin q_4 \left(\cos q_5 \frac{\partial \Psi}{\partial q_2} - \sin q_5 \frac{\partial \Psi}{\partial q_1} \right) \right), 0 \right) \quad (4) \end{aligned}$$

A and B are Lie operators, or vector fields

$$A = \frac{\partial H}{\partial \mathbf{p}} \cdot \frac{\partial}{\partial \mathbf{q}} \quad B = - \frac{\partial H}{\partial \mathbf{q}} \cdot \frac{\partial}{\partial \mathbf{p}} \quad (5)$$

The transfer to the operators are given in the following description.

The exponential operators e^{hA} and e^{hB} are then just shift operators, with $\mathcal{T}_2(h)$ is a symmetric second order splitting method:

$$\mathcal{T}_{2,VV}(h)(\Delta t) = e^{(\Delta t/2)B} e^{\Delta t A} e^{(\Delta t/2)B}. \quad (6)$$

and corresponds to the velocity form of the Verlet algorithm (VV).

Further the splitting scheme:

$$\mathcal{T}_{2,PV}(h)(\Delta t) = e^{(\Delta t/2)A} e^{\Delta t B} e^{(\Delta t/2)A}. \quad (7)$$

and corresponds to the position-form of the Verlet algorithm (PV).

See also the derivation of the Verlet algorithm in Appendix 7.

$\mathcal{T}_{2,VV}(h)(\Delta t) = S_{AB}(h)$, the symplectic Verlet or leap-frog algorithm is given as:

We start with $(\mathbf{q}_0, \mathbf{p}_0)^t = (\mathbf{q}(t^n), \mathbf{p}(t^n))^t$:

$$(\mathbf{q}_1, \mathbf{p}_1)^t = e^{h/2B}(\mathbf{q}_0, \mathbf{p}_0)^t = \left(I - \frac{1}{2}h \sum_i \frac{\partial H}{\partial \mathbf{q}}(\mathbf{p}_i, \mathbf{q}_i) \frac{\partial}{\partial \mathbf{p}_i}\right)(\mathbf{q}_0, \mathbf{p}_0)^t, \quad (8)$$

$$(\mathbf{q}_2, \mathbf{v}_2)^t = e^{hA}(\mathbf{q}_1, \mathbf{v}_1)^t = \left(I + h \sum_i \frac{\partial H}{\partial \mathbf{p}}(\mathbf{p}_i, \mathbf{q}_i) \frac{\partial}{\partial \mathbf{q}_i}\right)(\mathbf{q}_1, \mathbf{p}_1)^t, \quad (9)$$

$$(\mathbf{q}_3, \mathbf{v}_3)^t = e^{h/2B}(\mathbf{q}_2, \mathbf{p}_2)^t = \left(I - \frac{1}{2}h \sum_i \frac{\partial H}{\partial \mathbf{q}}(\mathbf{p}_i, \mathbf{q}_i) \frac{\partial}{\partial \mathbf{p}_i}\right)(\mathbf{q}_2, \mathbf{p}_2)^t. \quad (10)$$

And the substitution is given the algorithm for one time-step $n \rightarrow n+1$ and we obtain:

$$(\mathbf{q}(t^{n+1}), \mathbf{v}(t^{n+1}))^t = (\mathbf{q}_3, \mathbf{v}_3)^t.$$

1.2 Higher order Expansion of Verlet-algorithm

In the following we extend the Verlet algorithm with respect to higher order terms.

Such terms are important in the application with iterative schemes to achieve higher order schemes.

We start with $(\mathbf{q}_0, \mathbf{p}_0)^t = (\mathbf{q}(t^n), \mathbf{p}(t^n))^t$:

$$\begin{aligned} (\mathbf{q}_1, \mathbf{p}_1)^t &= e^{h/2B}(\mathbf{q}_0, \mathbf{p}_0)^t \\ &= \left(I + \sum_{j=1}^N \frac{1}{j!} \left(-\frac{1}{2}h \sum_i \frac{\partial H}{\partial \mathbf{q}}(\mathbf{p}_i, \mathbf{q}_i) \frac{\partial}{\partial \mathbf{p}_i}\right)^j\right)(\mathbf{q}_0, \mathbf{p}_0)^t, \end{aligned} \quad (11)$$

$$\begin{aligned}
(\mathbf{q}_2, \mathbf{v}_2)^t &= e^{hA}(\mathbf{q}_1, \mathbf{v}_1)^t \\
&= (I + \sum_{j=1}^N \frac{1}{j!} (h \sum_i \frac{\partial H}{\partial \mathbf{p}}(\mathbf{p}_i, \mathbf{q}_i) \frac{\partial}{\partial \mathbf{q}_i})^j)(\mathbf{q}_0, \mathbf{p}_0)^t, \quad (12)
\end{aligned}$$

$$\begin{aligned}
(\mathbf{q}_3, \mathbf{v}_3)^t &= e^{h/2B}(\mathbf{q}_2, \mathbf{p}_2)^t \\
&= (I + \sum_{j=1}^N \frac{1}{j!} (-\frac{1}{2}h \sum_i \frac{\partial H}{\partial \mathbf{q}}(\mathbf{p}_i, \mathbf{q}_i) \frac{\partial}{\partial \mathbf{p}_i})^j)(\mathbf{q}_0, \mathbf{p}_0)^t. \quad (13)
\end{aligned}$$

And the substitution is given the algorithm for one time-step $n \rightarrow n+1$ and we obtain:

$$(\mathbf{q}(t^{n+1}), \mathbf{v}(t^{n+1}))^t = (\mathbf{q}_3, \mathbf{v}_3)^t.$$

2 Iterative Schemes for coupled problems

Based on the nonlinear equations, we have to deal with linearization or nonlinear averaging techniques. In the following, we discuss the fixed point iteration and Newton's method.

We solve the nonlinear problem:

$$F(x) = 0, \quad (14)$$

where $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$.

2.1 Fixed-point iteration

The nonlinear equations can be formulated as fixed-point problems:

$$x = K(x), \quad (15)$$

where K is the fixed-point map and is nonlinear, e.g. $K(x) = x - F(x)$.

A solution of (16) is called fix-point of the map K .

The fix-point iteration is given as:

$$x_{i+1} = K(x_i), \quad (16)$$

and is called *nonlinear Richardson iteration*, *Picard iteration*, or *the method of successive substitution*.

Definition 2.1 Let $\Omega \leq \mathbb{R}^n$ and let $G : \Omega \rightarrow \mathbb{R}^m$. G is Lipschitz continuous on Ω with Lipschitz constant γ if

$$\|G(x) - G(y)\| \leq \gamma \|x - y\|, \quad (17)$$

for all $x, y \in \Omega$.

For the convergence we have to assume that K be a contraction map on Ω with Lipschitz constant $\gamma < 1$.

Algorithm 2.1 *We apply the fix-point iterative scheme to decouple the non-separable Hamiltonian problem (3) and (4).*

$$\dot{\mathbf{q}}_i = \frac{\partial H}{\partial \mathbf{p}}(\mathbf{p}_{i-1}, \mathbf{q}_{i-1}), t \in [t^n, t^{n+1}] \quad (18)$$

$$\dot{\mathbf{p}}_i = -\frac{\partial H}{\partial \mathbf{q}}(\mathbf{p}_{i-1}, \mathbf{q}_{i-1}), t \in [t^n, t^{n+1}] \quad (19)$$

$$p(t^0) = p_0, q(t^0) = q_0, \quad (20)$$

the starting solutions for the i -th iterative steps are given as :

$\mathbf{p}_{i-1}(t), \mathbf{q}_{i-1}(t)$ are the solutions of the $i-1$ th iterative step

and we have the initial condition for the fix-point iteration:

$$(\mathbf{p}_0(t), \mathbf{q}_0(t))^t = (\mathbf{p}(t^n), \mathbf{q}(t^n))^t$$

We assume that we have convergent results after $i = 1 \dots, m$ iterative steps or with the stopping criterion:

$$\max(\|\mathbf{p}_{i+1} - \mathbf{p}_i\|, \|\mathbf{q}_{i+1} - \mathbf{q}_i\|) \leq \text{err},$$

while $\|\cdot\|$ is the Euclidean norm (or a simple vector-norm, e.g. L_2).

Iterative Verlet applied to the Hamiltonian (3) and (4) :

We start with $(\mathbf{q}_0, \mathbf{p}_0)^t = (\mathbf{q}(t^n), \mathbf{p}(t^n))^t$:

The iterative scheme is given as:

$$\mathbf{q}_i(t) = \mathbf{q}(t^n) + h \frac{\partial H}{\partial \mathbf{p}}(\mathbf{p}(t^n) - \frac{h}{2} \frac{\partial H}{\partial \mathbf{q}}(\mathbf{p}_{i-1}(t), \mathbf{q}_{i-1}(t)), \mathbf{q}(t^n)), \quad (21)$$

$$\begin{aligned} \mathbf{p}_i(t) = & \mathbf{p}(t^n) - \frac{h}{2} \frac{\partial H}{\partial \mathbf{q}}(\mathbf{p}_{i-1}(t), \mathbf{q}_{i-1}(t)) \\ & - \frac{h}{2} \frac{\partial H}{\partial \mathbf{q}} \left(\left(\mathbf{p}(t^n) - \frac{h}{2} \frac{\partial H}{\partial \mathbf{q}}(\mathbf{p}_{i-1}(t), \mathbf{q}_{i-1}(t)) \right), \right. \\ & \left. \mathbf{q}(t^n) + h \frac{\partial H}{\partial \mathbf{p}} \left(\mathbf{p}(t^n) - \frac{h}{2} \frac{\partial H}{\partial \mathbf{q}}(\mathbf{p}_{i-1}(t), \mathbf{q}_{i-1}(t)), \mathbf{q}(t^n) \right) \right), \end{aligned} \quad (22)$$

$$\text{for } t \in [t^n, t^{n+1}], h = t^{n+1} - t^n, n = 0, 1, \dots, N, \quad (23)$$

$$i = 1, 2, 3 \dots, I, \quad (24)$$

where $I = 3$ or 4 .

For the fix-point iteration, we have the problem of the initialization, means the start of the iterative scheme. We can improve the starting solution with a preprocessing method, which derives a first improved initial solution.

Improve Initialization Process

Algorithm 2.2 *To improve the initial solution we can start with:*

- 1.) *We initialize with a result of the explicit Euler-method :*
 $(\mathbf{q}_0, \mathbf{p}_0)^t = (\mathbf{q}(t^{n+1})_{Euler1st}, \mathbf{p}(t^{n+1})_{Euler1st})^t$;
- 2.) *We initialize with a result of the explicit RK-method :*
 $(\mathbf{q}_0, \mathbf{p}_0)^t = (\mathbf{q}(t^{n+1})_{RK4th}, \mathbf{p}(t^{n+1})_{RK4th})^t$;

2.2 Newton's method

We solve the nonlinear operator equation (14).

While $F : D \subset X \rightarrow Y$ with the Banach spaces X, Y is given with the norms $\|\cdot\|_X$ and $\|\cdot\|_Y$. Let F be at least once continuous differentiable, further we assume x_0 is a starting solution of the unknown solution x^* .

Then the *successive linearization* lead to the general Newton's method:

$$F'(x_i)\Delta x_i = -F(x_i), \quad (25)$$

where $\Delta x_i = x_{i+1} - x_i$ and $i = 0, 1, 2, \dots$.

The method derive the solution of a nonlinear problem by solving the following algorithm.

Algorithm 2.3 *By considering the sequential splitting method we obtain the following algorithm. We apply the equations*

$$\dot{\mathbf{q}} - \frac{\partial H}{\partial \mathbf{p}}(\mathbf{p}, \mathbf{q}) = 0, t \in [t^n, t^{n+1}] \quad (26)$$

$$\dot{\mathbf{p}} + \frac{\partial H}{\partial \mathbf{q}}(\mathbf{p}, \mathbf{q}) = 0, t \in [t^n, t^{n+1}] \quad (27)$$

$$\mathbf{p}_n = \mathbf{p}(t^n), \mathbf{q}_n = \mathbf{q}(t^n), \quad (28)$$

where $\mathbf{p} = (p_1, \dots, p_6)^t$ and $\mathbf{q} = (q_1, \dots, q_6)^t$ into the Newtons-formula we have:

$$F(\mathbf{p}, \mathbf{q}) = \dot{\mathbf{x}} + \frac{\partial H}{\partial \mathbf{x}}(\mathbf{x}) \quad (29)$$

and we can compute

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - D(F(\mathbf{x}^{(k)}))^{-1} F(\mathbf{x}^{(k)}), \quad (30)$$

where $D(F(\mathbf{x}))$ is the Jacobian matrix and $k = 0, 1, \dots$.

We stop the iterations when we obtain : $|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}| \leq \text{err}$, where *err* is an error bound, e.g. *err* = 10^{-4} .

The solution vector F is given as:

$$F(\mathbf{x}) = \begin{pmatrix} F_{q,1}(\mathbf{x}) \\ F_{q,2}(\mathbf{x}) \\ \vdots \\ F_{q,6}(\mathbf{x}) \\ F_{p,1}(\mathbf{x}) \\ F_{p,2}(\mathbf{x}) \\ \vdots \\ F_{p,6}(\mathbf{x}) \end{pmatrix} \quad (31)$$

where $\mathbf{x} = (q_1, \dots, q_6, p_1, \dots, p_6)^t$ and

$$F(\mathbf{x}) = \begin{pmatrix} F_q(\mathbf{x}) \\ F_p(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} \dot{\mathbf{q}} - \frac{\partial H}{\partial \mathbf{p}} \\ \dot{\mathbf{p}} - \frac{\partial H}{\partial \mathbf{q}} \end{pmatrix}. \quad (32)$$

The Jacobian matrix for the equation system is given as :

$$DF(\mathbf{x}) = \begin{pmatrix} \frac{\partial F_1}{\partial x_1} & \frac{\partial F_1}{x_2} & \cdots & \frac{\partial F_1}{\partial x_{12}} \\ \frac{\partial F_2}{\partial x_1} & \frac{\partial F_2}{x_2} & \cdots & \frac{\partial F_2}{\partial x_{12}} \\ \vdots & & & \\ \frac{\partial F_{12}}{\partial x_1} & \frac{\partial F_{12}}{x_2} & \cdots & \frac{\partial F_{12}}{\partial x_{12}} \end{pmatrix} \quad (33)$$

where $\mathbf{x} = (x_1, \dots, x_{12})^t = (q_1, \dots, q_6, p_1, \dots, p_6)^t$.

3 Splitting Methods

In the following, we discuss the different splitting schemes.

The simplest such symmetric product is

$$\mathcal{T}_2(h) = S_{AB}(h) \quad \text{or} \quad \mathcal{T}_2(h) = S_{BA}(h). \quad (34)$$

If one naively assumes that

$$\mathcal{T}_2(h) = e^{\Delta t(A+B)} + Ch^3 + Dh^4 + \dots, \quad (35)$$

then a Richardson extrapolation would only give

$$\frac{1}{k^2 - 1} [k^2 \mathcal{T}_2^k(h/k) - \mathcal{T}_2(h)] = e^{\Delta t(A+B)} + O(h^4), \quad (36)$$

a third-order algorithm.

Thus for a given set of n whole numbers $\{k_i\}$ one can have a $2n$ th-order approximation

$$e^{\Delta t(A+B)} = \sum_{i=1}^n c_i \mathcal{T}_2^{k_i} \left(\frac{\Delta t}{k_i} \right) + O(h^{2n+1}). \quad (37)$$

For orders four to ten, one has explicitly:

$$\mathcal{T}_4(\Delta t) = -\frac{1}{3}\mathcal{T}_2(\Delta t) + \frac{4}{3}\mathcal{T}_2^2 \left(\frac{\Delta t}{2} \right), \quad (38)$$

$$\mathcal{T}_6(\Delta t) = \frac{1}{24}\mathcal{T}_2(\Delta t) - \frac{16}{15}\mathcal{T}_2^2 \left(\frac{\Delta t}{2} \right) + \frac{81}{40}\mathcal{T}_2^3 \left(\frac{\Delta t}{3} \right), \quad (39)$$

$$\mathcal{T}_8(\Delta t) = -\frac{1}{360}\mathcal{T}_2(\Delta t) + \frac{16}{45}\mathcal{T}_2^2 \left(\frac{\Delta t}{2} \right) - \frac{729}{280}\mathcal{T}_2^3 \left(\frac{\Delta t}{3} \right) + \frac{1024}{315}\mathcal{T}_2^4 \left(\frac{\Delta t}{4} \right), \quad (40)$$

$$\begin{aligned} \mathcal{T}_{10}(\Delta t) = & \frac{1}{8640}\mathcal{T}_2(\Delta t) - \frac{64}{945}\mathcal{T}_2^2 \left(\frac{\Delta t}{2} \right) + \frac{6561}{4480}\mathcal{T}_2^3 \left(\frac{\Delta t}{3} \right) \\ & - \frac{16384}{2835}\mathcal{T}_2^4 \left(\frac{\Delta t}{4} \right) + \frac{390625}{72576}\mathcal{T}_2^5 \left(\frac{\Delta t}{5} \right). \end{aligned} \quad (41)$$

4 Iterative MPE method

Based on the nonlinear problem, we extend the MPE method to an iterative scheme.

Algorithm 4.1 *Iterative Verlet applied to the Hamiltonian (3) and (4) :*

We start with $(\mathbf{q}_0, \mathbf{p}_0)^t = (\mathbf{q}(t^n), \mathbf{p}(t^n))^t$:

The iterative scheme for computing $\mathcal{T}_2(p_i, q_i, h) = (\mathbf{q}_i(h), \mathbf{p}_i(h)^t)$ is given as:

$$\begin{aligned} \mathcal{T}_2(p_i, q_i, h) &= e^{h/2B} e^{hA} e^{h/2B} (\mathbf{q}_0, \mathbf{p}_0)^t \\ &= \left(I + \sum_{j=1}^N \frac{1}{j!} \left(-\frac{1}{2} h \frac{\partial H}{\partial \mathbf{q}}(\mathbf{p}_{i-1}, \mathbf{q}_{i-1}) \frac{\partial}{\partial \mathbf{p}(t^n)} \right)^j \right) \\ &\quad \left(I + \sum_{j=1}^N \frac{1}{j!} \left(h \frac{\partial H}{\partial \mathbf{p}}(\mathbf{p}_{i-1}, \mathbf{q}_{i-1}) \frac{\partial}{\partial \mathbf{q}(t^n)} \right)^j \right) \\ &\quad \left(I + \sum_{j=1}^N \frac{1}{j!} \left(-\frac{1}{2} h \frac{\partial H}{\partial \mathbf{q}}(\mathbf{p}_{i-1}, \mathbf{q}_{i-1}) \frac{\partial}{\partial \mathbf{p}(t^n)} \right)^j \right) (\mathbf{q}(t^n), \mathbf{p}(t^n))^t \end{aligned} \quad (42)$$

We have the higher order schemes given as:

$$\mathcal{T}_4(p_i, q_i, h) = -\frac{1}{3}\mathcal{T}_2(p_i, q_i, h) + \frac{4}{3}\mathcal{T}_2^2\left(p_i, q_i, \frac{h}{2}\right) \quad (43)$$

$$\mathcal{T}_6(p_i, q_i, h) = \frac{1}{24}\mathcal{T}_2(p_i, q_i, h) - \frac{16}{15}\mathcal{T}_2^2\left(p_i, q_i, \frac{h}{2}\right) + \frac{81}{40}\mathcal{T}_2^3\left(p_i, q_i, \frac{h}{3}\right) \quad (44)$$

$$\begin{aligned} \mathcal{T}_8(p_i, q_i, h) = & -\frac{1}{360}\mathcal{T}_2(p_i, q_i, h) + \frac{16}{45}\mathcal{T}_2^2\left(p_i, q_i, \frac{h}{2}\right) \\ & - \frac{729}{280}\mathcal{T}_2^3\left(p_i, q_i, \frac{h}{3}\right) + \frac{1024}{315}\mathcal{T}_2^4\left(p_i, q_i, \frac{h}{4}\right) \end{aligned} \quad (45)$$

$$\begin{aligned} \mathcal{T}_{10}(p_i, q_i, h) = & \frac{1}{8640}\mathcal{T}_2(p_i, q_i, h) - \frac{64}{945}\mathcal{T}_2^2\left(p_i, q_i, \frac{h}{2}\right) + \frac{6561}{4480}\mathcal{T}_2^3\left(p_i, q_i, \frac{h}{3}\right) \\ & - \frac{16384}{2835}\mathcal{T}_2^4\left(p_i, q_i, \frac{h}{4}\right) + \frac{390625}{72576}\mathcal{T}_2^5\left(p_i, q_i, \frac{h}{5}\right). \end{aligned} \quad (46)$$

where for $t \in [t^n, t^{n+1}]$, $h = t^{n+1} - t^n$, $n = 0, 1, \dots, N$,
 $i = 1, 2, 3 \dots, I$,

and we have a stopping criterion or a fixed number of iterative steps, e.g.
 $I = 3$ or 4.

5 Numerical Examples

The Levitron is described on the base of rigid body theory. With the convention of Goldstein [9] for the Euler angles the angular velocity ω_ϕ is along the z -axis of the system, ω_θ along the line of nodes and ω_ψ along the z' -axis. Transforming them into body coordinates one gets

$$\omega = \begin{pmatrix} \dot{\phi} \sin \theta \sin \psi + \dot{\theta} \cos \psi \\ \dot{\phi} \sin \theta \cos \psi + \dot{\theta} \sin \psi \\ \dot{\phi} \cos \theta + \dot{\psi} \end{pmatrix} \quad (47)$$

Finally the kinetic energy can be written as

$$T = \frac{1}{2} \left[m(\dot{x}^2 + \dot{y}^2 + \dot{z}^2) + A(\dot{\theta}^2 + \dot{\phi}^2 \sin^2 \theta) + C(\dot{\psi} + \dot{\phi} \cos \theta)^2 \right] \quad (48)$$

The potential energy U is given by the sum of the gravitational energy and the interaction potential of the Levitron in the magnetic field of the base plate:

$$U = mgz - \mu(\sin \psi \sin \theta \frac{\Phi}{x} + \cos \psi \sin \theta \frac{\Phi}{y} + \cos \theta \frac{\Phi}{z}) \quad (49)$$

with μ as the magnetic moment of the top and Φ the magneto-static potential. Following Gans [6] we use the potential of a ring dipole as approximation for a magnetized plane with a centered unmagnetized hole. Furthermore we

introduced a nondimensionalization for the variables and the magneto-static potential:

$$\Psi = \frac{Z}{(1+Z^2)^{3/2}} - (X^2 + Y^2) \frac{3(2Z^2 - 3)Z}{4(1+Z^2)^{7/2}} \quad (50)$$

Lengths were scaled by the radius R of the base plane, mass were measured in units of m and energy in units of mgh . Therefore the one time unit is $\sqrt{R/g}$. So the dimensionless Hamiltonian with $\mathbf{q} = (\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \theta, \psi, \phi)$ is given by

$$H = \frac{1}{2} \left(p_1^2 + p_2^2 + p_3^2 + \frac{p_4^2}{a} + \frac{(p_5 - P - 6 \cos q_4)^2}{a \sin^2 q_4} + \frac{p_6^2}{c} \right) \quad (51)$$

$$- M \left[\sin q_4 \left(\cos q_5 \frac{\partial \Psi}{\partial q_1} \sin q_5 \frac{\partial \Psi}{\partial q_2} \right) + \cos q_4 \frac{\partial \Psi}{\partial q_3} \right] + q_3 \quad (52)$$

with a and c as the nondimensionalized inertial parameter and M as the ratio of gravitational and magnetic energy.

Solving the equations of motion (3) and (4) numerically with the methods described above, one is able to plot the movement of the center of mass like it is shown in figure 1. In the plot the first 5 seconds of a stable trajectory were plotted. With a longer calculation we tested, that the top would levitate for more than one minute.

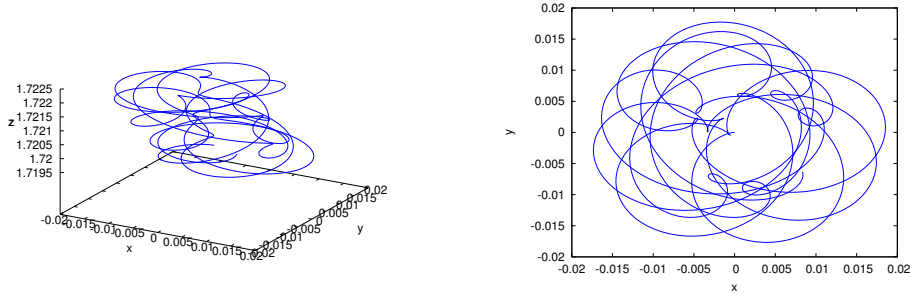


Figure 1: stable trajectory of the center of mass in 3D and 2D

The axis in the plot show the nondimensional variables X, Y and Z . The trajectory starts at the equilibrium point $(q_1, q_2, q_3) = (0, 0, 1.72)$. This trajectory was calculated with the fourth-order RungeKutta method with a small timestep of 10^{-5} units of time, where one time unit is about 59,5ms, because of the nondimensionalization. For further considerations this trajectory will be used as a reference solution.

The errors of the different time-steps with Runge-Kutta are given in Figure 2.

The same equations were solved with the iterative Verlet algorithm described before. Due to the long computation time needed, we simulated only 1000 timesteps and compare the trajectory with the reference solution from the

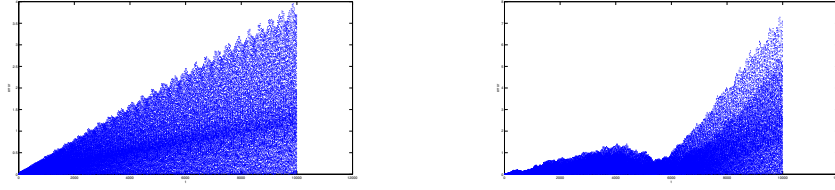


Figure 2: Errors of the numerical scheme: Runge-Kutta method (explicit 4th order).

Runge-Kutta algorithm. In figure 3 is shown how the trajectory of the same initial conditions looks like with the Verlet algorithm.

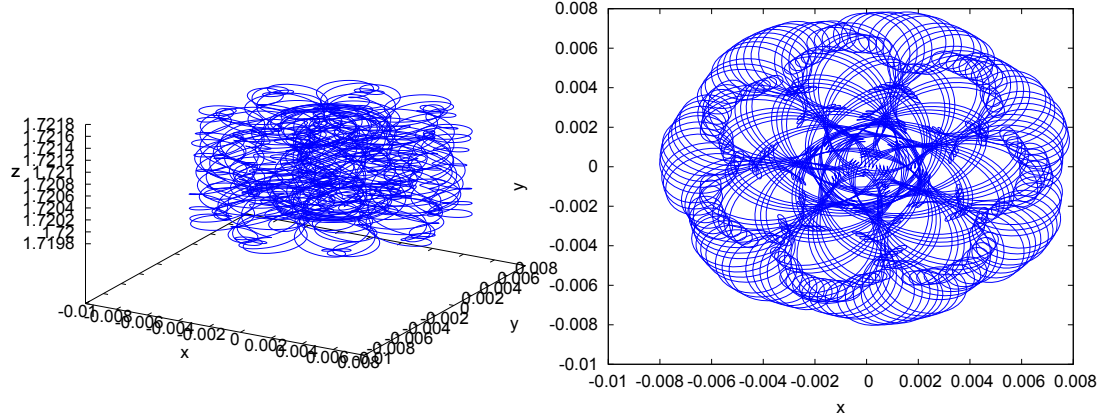


Figure 3: trajectory calculated with Verlet algorithm

This were done for one, two and four iterations per timestep, to see whether how many iterations are reasonable. The results are shown in Figure 4.

In a first comparison, we deal with the second order Verlet algorithm and improve the scheme with iterative steps.

In a first initialisation process, one can see that the errors are very similar to 1 or 2 iterative steps.

The reduction of the error is possible with the improvement to higher order initialisation scheme, e.g. start with a first approximate solution with a RK scheme, or apply extrapolation schemes.

The following tables 1 and 2 should give an impression of the timescales of the problem and the errors.

Remark 5.1 Obviously the iterations does not improve the algorithm, when only using a lower order initialisation. By the way, it is sufficient to apply one iterative step in in comparison with the Runge-Kutta algorithm.

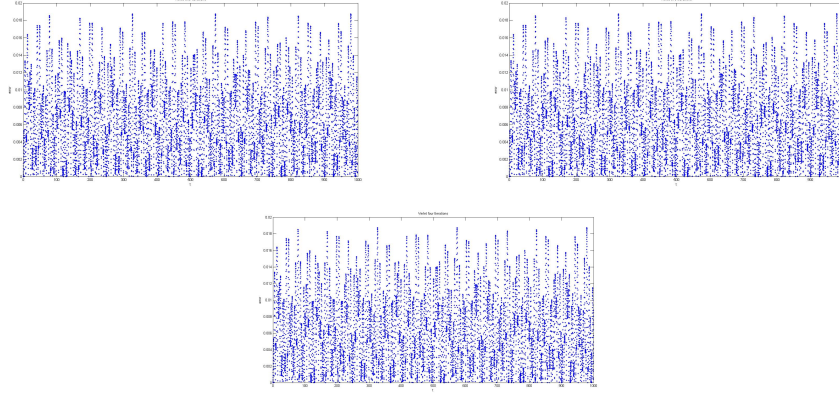


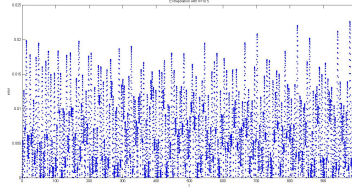
Figure 4: Errors of the numerical scheme: Iterative Verlet method.

Runge-Kutta			
timestep	10^{-5}	10^{-3}	10^{-1}
number of steps	1000000000	10000000	100000
computing time	119 min	23 sec	2 sec
stability	ok	ok	ok

Table 1: Stability and Computational Time with 4th order explicit Runge-Kutta.

Also we have a benefit in reducing the computational time instead of applying only Runge-Kutta schemes.

We tried to improve the solution with a extrapolation scheme in fourth order. We have a view at the errors this algorithm produces in comparison with the Runge-Kutta Solution with small time-steps (10^{-5} time units per step). In Figure 5, we presented the results of the 4th MPE method with different time-steps and compared it with the Runge-Kutta solution.

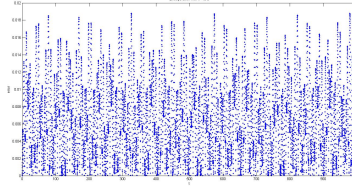
Figure 5: Errors of the numerical scheme: 4th order Extrapolation Scheme with Verlet method a Kernel ($h = 10^{-5}$).

Also we tested the 6th order MPE method with different time-steps and

Verlet			
timestep	10^{-6}	10^{-6}	10^{-6}
iterations per step	1	2	4
stability	ok	ok	ok
computing time	67min	120min	219min
mean error	0.068	0.068	0.068
maximal error	0.0187	0.0188	0.0187

Table 2: Stability and Computational Time with 2nd order Verlet Scheme.

compared it with the Runge-Kutta solution, see Figure 6.

Figure 6: Errors of the numerical scheme: 6th order Extrapolation Scheme with Verlet method a Kernel ($h = 10^{-5}$).

Like for Runge-Kutta we want to give an impression of the time scales for this extrapolation schemes, see Table 3.

	Extrapolation 4th order		Extrapolation 6th order	
	10^{-5}	10^{-6}	10^{-5}	10^{-6}
number of steps	100000000	1000000000	100000000	1000000000
computing time	14min	142min	29min	272min
mean error	0.007	0.007	0.0068	0.0068
maximal error	0.0226	0.0234	0.0188	0.0188

Table 3: Errors and Computational Time with 4th order MPE scheme using Verlet Scheme as Kernel.

6 Conclusions and Discussions

In the paper, we have presented a model to simulate a Levitron. Based on the given Hamiltonian system, which is nonlinear, we present novel and simpler schemes based on splitting ideas to solve the equation systems. In future, we concentrate on the numerical analysis and embedding higher order splitting kernels to the extrapolation schemes.

7 Appendix

For example, the evolution of any dynamical variable $u(\mathbf{q}, \mathbf{p})$ (including \mathbf{q} and \mathbf{p} themselves) is given by the Poisson bracket,

$$\partial_t u(\mathbf{q}, \mathbf{p}) = \left(\frac{\partial u}{\partial \mathbf{q}} \cdot \frac{\partial H}{\partial \mathbf{p}} - \frac{\partial u}{\partial \mathbf{p}} \cdot \frac{\partial H}{\partial \mathbf{q}} \right) = (A + B)u(\mathbf{q}, \mathbf{p}). \quad (53)$$

For a separable Hamiltonian,

$$H(\mathbf{p}, \mathbf{q}) = \frac{\mathbf{p}^2}{2m} + V(\mathbf{q}), \quad (54)$$

A and B are Lie operators, or vector fields

$$A = \mathbf{v} \cdot \frac{\partial}{\partial \mathbf{q}} \quad B = \mathbf{a}(\mathbf{q}) \cdot \frac{\partial}{\partial \mathbf{v}} \quad (55)$$

where we have abbreviated $\frac{\partial H}{\partial \mathbf{p}}(\mathbf{p}, \mathbf{q}) = \mathbf{v} = \mathbf{p}/m$ and $-\frac{\partial H}{\partial \mathbf{q}}(\mathbf{p}, \mathbf{q}) = \mathbf{a}(\mathbf{q}) = -\nabla V(\mathbf{q})/m$. The exponential operators e^{hA} and e^{hB} are then just shift operators.

$$S(h) = e^{h/2B} e^{hA} e^{h/2B}$$

That is also given as a Verlet-algorithm in the following scheme.

We start with $(\mathbf{q}_0, \mathbf{v}_0)^t = (\mathbf{q}(t^n), \mathbf{v}(t^n))^t$:

$$(\mathbf{q}_1, \mathbf{v}_1)^t = e^{h/2B}(\mathbf{q}_0, \mathbf{v}_0)^t = \left(I + \frac{1}{2}h \sum_i a(\mathbf{q}) \frac{\partial}{\partial \mathbf{v}_i} \right) (\mathbf{q}_0, \mathbf{v}_0)^t \quad (56)$$

$$= (\mathbf{q}_0, \mathbf{v}_0 + \frac{1}{2}ha(\mathbf{q}_0))^t, \quad (57)$$

$$(\mathbf{q}_2, \mathbf{v}_2)^t = e^{hA}(\mathbf{q}_1, \mathbf{v}_1)^t = \left(I + h \sum_i \mathbf{v}_i \frac{\partial}{\partial \mathbf{q}_i} \right) (\mathbf{q}_1, \mathbf{v}_1)^t \quad (58)$$

$$= (\mathbf{q}_1 + h\mathbf{v}_1, \mathbf{v}_1)^t, \quad (59)$$

$$(\mathbf{q}_3, \mathbf{v}_3)^t = e^{h/2B}(\mathbf{q}_2, \mathbf{v}_2)^t = \left(I + \frac{1}{2}h \sum_i a(\mathbf{q}) \frac{\partial}{\partial \mathbf{v}_i} \right) (\mathbf{q}_2, \mathbf{v}_2)^t \quad (60)$$

$$= (\mathbf{q}_2, \mathbf{v}_2 + \frac{1}{2}ha(\mathbf{q}_1))^t. \quad (61)$$

And the substitution is given the algorithm for one time-step $n \rightarrow n+1$:

$$(\mathbf{q}_3, \mathbf{v}_3)^t = (\mathbf{q}_0 + h\mathbf{v}_0 + \frac{h^2}{2}a(\mathbf{q}_0), \mathbf{v}_0 + \frac{h}{2}a(\mathbf{q}_0) + \frac{h}{2}a(\mathbf{q}_0 + h\mathbf{v}_0 + \frac{h}{2}a(\mathbf{q}_0)))^t, \quad (62)$$

while $(\mathbf{q}(t^{n+1}), \mathbf{v}(t^{n+1}))^t = (\mathbf{q}_3, \mathbf{v}_3)^t$.

Iterative Verlet Algorithm

In the abstract version of $\frac{\partial H}{\partial \mathbf{p}}(\mathbf{p}_{i-1}, \mathbf{q}_{i-1}q)$, $-\frac{\partial H}{\partial \mathbf{q}}(\mathbf{p}_{i-1}, \mathbf{q}_{i-1})$.

Algorithm 7.1 *We have the iterative Verlet Algorithm:*

- 1.) *We start with the initialisation : $(\mathbf{p}_0(t^{n+1}), \mathbf{q}_0(t^{n+1}))^t = (\mathbf{p}(t^n), \mathbf{q}(t^n))^t$ and $i = 0$*
- 2.) *The iterative step is given as: $i = i + 1$ and we have:*

$$\mathbf{q}_i^{n+1} = \mathbf{q}^n + h \frac{\partial H}{\partial \mathbf{p}}(\mathbf{p}^n - \frac{1}{2} h \frac{\partial H}{\partial \mathbf{q}}(\mathbf{p}_{i-1}^{n+1}, \mathbf{q}_{i-1}^{n+1}), \mathbf{q}_{i-1}^{n+1}), \quad (63)$$

$$\mathbf{p}_i(t^{n+1}) = \mathbf{p}^n - \frac{h}{2} \frac{\partial H}{\partial \mathbf{q}}(\mathbf{p}_{i-1}^{n+1}, \mathbf{q}_{i-1}^{n+1}), \quad (64)$$

$$\mathbf{p}_i(t^{n+1}) = \mathbf{p}^n - \frac{h}{2} \frac{\partial H}{\partial \mathbf{q}}(\mathbf{p}_{i-1}^{n+1}, \mathbf{q}^n + h \frac{\partial H}{\partial \mathbf{p}}(\mathbf{p}^n - \frac{1}{2} h \frac{\partial H}{\partial \mathbf{q}}(\mathbf{p}_{i-1}^{n+1}, \mathbf{q}_{i-1}^{n+1}), \mathbf{q}_{i-1}^{n+1})),$$

where $h = t^{n+1} - t^n$ is the local time-step.

We compute the stopping criterion:

$\max(\|\mathbf{p}(t_i^{n+1}) - \mathbf{p}_{i-1}^{n+1}\|, \|\mathbf{q}(t_i^{n+1}) - \mathbf{q}_{i-1}^{n+1}\|) \leq \text{err}$ or we stop after $i = I$, while I is the maximal iterative step.

3.) *The result is given as:*

$$(\mathbf{p}(t^{n+1}), \mathbf{q}(t^{n+1}))^t = (\mathbf{p}_i(t^{n+1}), \mathbf{q}_i(t^{n+1}))^t$$

and $n = n + 1$ if $n > N$, while N is the maximal time-step, we stop else we go to step 1.)

References

- [1] S. Chin and J. Geiser. *Multi-product operator splitting as a general method of solving autonomous and non-autonomous equations*. IMA J. Numer. Anal., first published online January 12, 2011.
- [2] B. Davis. *Integral Transform and Their Applications*. Applied Mathematical Sciences, 25, Springer Verlag, New York, Heidelberg, Berlin, 1978 .
- [3] H.R. Dullin and R. Easton. *Stability of Levitron*. Physica D: Nonlinear Phenomena, vol. 126, no. 1-2, 1-17, 1999.
- [4] K.-J. Engel, R. Nagel, *One-Parameter Semigroups for Linear Evolution Equations*. Springer-Verlag, Heidelberg, New York, 2000.
- [5] I. Farago and J. Geiser. *Iterative Operator-Splitting Methods for Linear Problems*. Preprint No. 1043 of the Weierstrass Institute for Applied Analysis and Stochastics, (2005) 1-18. International Journal of Computational Science and Engineering, accepted September 2007.

- [6] R.F. Gans, T.B. Jones, and M. Washizu. *Dynamics of the Levitron*. J. Phys. D., 31, 671-679, 1998.
- [7] J. Geiser. *Higher order splitting methods for differential equations: Theory and applications of a fourth order method*. Numerical Mathematics: Theory, Methods and Applications. Global Science Press, Hong Kong, China, accepted, April 2008.
- [8] J. Geiser and L. Noack. *Iterative operator-splitting methods for nonlinear differential equations and applications of deposition processes* Preprint 2008-4, Humboldt University of Berlin, Department of Mathematics, Germany, 2008.
- [9] H. Goldstein, Ch.P. Poole, and J. Safko. *Classical mechanics*. Addison Wesley, San Francisco, USA, 2002.
- [10] M. Hieber, A. Holderrieth and F. Neubrander. *Regularized semigroups and systems of linear partial differential equations*. Annali della Scuola Normale Superiore di Pisa - Classe di Scienze, Ser.4, 19:3, 363-379, 1992.
- [11] M. Hochbruck and A. Ostermann. *Explicit Exponential Runge-Kutta Methods for Semilinear Parabolic Problems*. SIAM Journal on Numerical Analysis, 43:3, 1069-1090, 2005.
- [12] E. Hansen and A. Ostermann. *Exponential splitting for unbounded operators*. Mathematics of Computation, accepted, 2008.
- [13] Hildebrand, F.B. (1987) *Introduction to Numerical Analysis*. Second Edition, Dover Edition.
- [14] T. Jahnke and C. Lubich. *Error bounds for exponential operator splittings*. BIT Numerical Mathematics, 40:4, 735-745, 2000.
- [15] J. Kanney, C. Miller and C. Kelley. *Convergence of iterative split-operator approaches for approximating nonlinear reactive transport problems*. Advances in Water Resources, 26:247-261, 2003.
- [16] C.T. Kelly. *Iterative Methods for Linear and Nonlinear Equations*. Frontiers in Applied Mathematics, SIAM, Philadelphia, USA, 1995.
- [17] G.I. Marchuk. *Some applications of splitting-up methods to the solution of problems in mathematical physics*. Aplikace Matematiky, 1, 103-132, 1968.
- [18] G. Strang. *On the construction and comparison of difference schemes*. SIAM J. Numer. Anal., 5, 506-517, 1968.
- [19] I. Najfeld and T.F. Havel. *Derivatives of the matrix exponential and their computation*. Adv. Appl. Math, ftp://ftp.das.harvard.edu/pub/cheatham/tr-33-94.ps.gz, 1995.

- [20] Strang, G. (1968) On the construction and comparison of difference schemes. *SIAM J. Numer. Anal.*, **5**, 506-517.
- [21] M. Suzuki. *General Decomposition Theory of Ordered Exponentials*. Proc. Japan Acad., 69, Ser. B, 161, 1993.
- [22] H. Goldstein. *Klassische Mechanik*. Akademische Verlagsgesellschaft, Wiesbaden, 1981.